

**JP05016450 A**  
**DATA OUTPUT APPARATUS**  
**RICOH CO LTD**

**Abstract:**

**PURPOSE:** To enhance the use efficiency of the whole of an apparatus by continuously storing residual font data in an auxiliary memory device when all of font data can not be stored in a memory because the quantity of the font data subjected to down load from a host device is much with respect to the capacity of the memory.

**CONSTITUTION:** A laser printer 2 is equipped with a printer controller 4 analyzing printing data such as character data or image data sent from a host computer 1 to convert the same to bit map data in a page unit. In a printer engine 5, laser is modulated corresponding to the bit map data to print a visible image on printing paper. In this case, a readable and writable auxiliary memory device 3 is connected to the printer controller 4 and, when the quantity of font data subjected to down load from a host device is much with respect to the capacity of the memory in the controller 4 and all of font data can not be stored in the memory, residual font data are stored in the auxiliary memory device 3.

**COPYRIGHT:** (C)1993,JPO&Japio

**Inventor(s):**

MORISHITA KOICHI  
OGAWA SATOSHI

**Application No.** 03193464 JP03193464 JP, **Filed** 19910708, **A1 Published** 19930126

**Original IPC(1-7):** B41J00530  
G06F00312

**Patents Citing This One** No US, EP, or WO patent/search reports have cited this patent.

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平5-16450

(43) 公開日 平成5年(1993)1月26日

(51) Int.Cl.<sup>5</sup>

B 4 1 J 5/30

G 0 6 F 3/12

識別記号

Z 8907-2C

G 8323-5B

庁内整理番号

F I

技術表示箇所

審査請求 未請求 請求項の数 3 (全 12 頁)

(21) 出願番号 特願平3-193464

(22) 出願日 平成3年(1991)7月8日

(71) 出願人 000006747

株式会社リコー

東京都大田区中馬込1丁目3番6号

(72) 発明者 森下 幸一

東京都大田区中馬込1丁目3番6号 株式

会社リコー内

(72) 発明者 小川 聡

東京都大田区中馬込1丁目3番6号 株式

会社リコー内

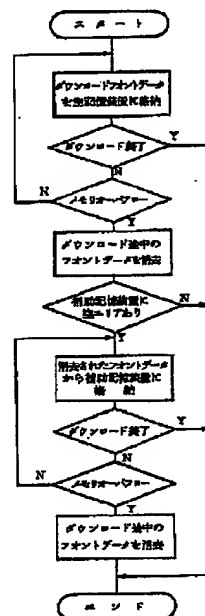
(74) 代理人 弁理士 大澤 敬

(54) 【発明の名称】 情報出力装置

(57) 【要約】

【目的】 上位装置よりフォントデータをダウンロードする機会を減らして、装置全体の使用効率を向上させる。

【構成】 上位装置であるホストコンピュータよりフォントデータがダウンロードされると、それを主記憶装置（装置内のRAM）に順次格納していくが、そのダウンロードが終了する前に主記憶装置のメモリ容量不足が生じ、メモリオーバーフローを検出した場合には、ホストコンピュータにダウンロードを中断させてその途中のフォントデータを消去する。次いで、補助記憶装置のメモリ容量をチェックし、そこに空きエリアがあれば、ホストコンピュータにダウンロード途中で消去されたフォントデータの先頭より再びダウンロードを開始させ、それを補助記憶装置に格納する。



1

## 【特許請求の範囲】

【請求項1】 上位装置よりフォントデータをダウンロードして装置内のメモリに格納し、そのフォントデータを用いて文字あるいは図形等の情報を出力する情報出力装置において、

読み書き可能な補助記憶装置と、前記メモリの容量に対して前記上位装置よりダウンロードするフォントデータ量が多く、全てのフォントデータを前記メモリに格納できない場合には、残りのフォントデータを継続して前記補助記憶装置に格納する手段とを設けたことを特徴とする情報出力装置。

【請求項2】 複数の情報出力装置をエミュレートする手段を有する情報出力装置において、

上位装置よりフォントデータをダウンロードする手段と、エミュレート可能な情報出力装置の間でダウンロードされたフォントデータを共有化する手段と、該手段によって共有化されたフォントデータを用いてエミュレートされた情報出力装置の種類に係らず文字イメージとしてビットマップメモリ上に展開する手段と、共有化されたフォントデータが存在するか否かを判断する手段と、該手段によって共有化されたフォントデータが存在すると判断された場合には、該フォントデータが格納されたメモリエリアをエミュレーションの切り替えが行なわれても保持する手段とを設けたことを特徴とする情報出力装置。

【請求項3】 請求項2記載の情報出力装置において、共有化されたフォントデータが格納されたメモリエリアをエミュレーションの切り替えが行なわれた場合に保持するか否かを選択する手段を設けたことを特徴とする情報出力装置。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】 この発明は、レーザプリンタ等のページプリンタ、デジタル複写機、普通紙ファックス装置等の画像形成装置やCRTディスプレイ装置などの情報出力装置に関する。

## 【0002】

【従来の技術】 レーザプリンタ等の情報出力装置において、上位装置（ホストコンピュータ）よりダウンロードされたフォントデータはビットマップ形式に展開され、装置内のメモリにフォントファイルとして格納される。このような情報出力装置においては、ダウンロードされるフォントデータ量が多くメモリ容量が不足すると、メモリオーバフローとなってダウンロード途中のフォントデータを削除してダウンロードを終了しなければならない。

【0003】 そこで、従来はメモリ容量よりも多くのフォントデータを必要とする場合には、新たにメモリを増設してメモリ容量を拡張したり、あるいはダウンロードしていないフォントデータが必要になった時に、メモリ

2

内の他のフォントデータを削除した後、必要とするフォントデータをダウンロードし、それをビットマップ形式に展開してメモリの空きエリアに格納することが行なわれている。

【0004】 一方、情報出力装置であるプリンタは各々独自のコマンド体系を持っているが、現在はプリンタの中には独自のコマンド体系に加えて他のプリンタのコマンド体系をエミュレートするプリンタが増加しつつある。各々のコマンド体系の中では各々の方法でフォントデータのダウンロードが行なわれており、エミュレーションが切り替えられた場合、それまで使用されていたダウンロードフォントデータはRAM上でクリアされ、新たに切り替えられたエミュレーションのための初期化がRAM上で行なわれる。

## 【0005】

【発明が解決しようとする課題】 しかしながら、前者の情報出力装置においては、メモリを増設するとその分だけ装置全体のコストが上昇するという問題があった。また、少ないメモリ容量で多くのダウンロードフォントを使用する場合、必要なフォントデータがダウンロードされていなければ、上位装置との印字データの通信を一時中断してダウンロードを行なう必要があり、処理に時間がかかるという問題があった。

【0006】 一方、後者のプリンタにおいては、エミュレーションの切り替えによってRAM上のダウンロードフォントが消去されてしまうため、再び切り替え前のエミュレーションに戻した場合でも、そのフォントデータは再びダウンロードしない限り使用することはできなかった。

【0007】 この発明は上記の点に鑑みてなされたものであり、上述のような問題を解消し、上位装置よりフォントデータをダウンロードする機会を極力減らして、装置全体の使用効率を向上させることを目的とする。

## 【0008】

【課題を解決するための手段】 この発明は上記の目的を達成するため、上位装置よりフォントデータをダウンロードして装置内のメモリに格納し、そのフォントデータを用いて文字あるいは図形等の情報を出力する情報出力装置において、読み書き可能な補助記憶装置と、上記メモリの容量に対して上位装置よりダウンロードするフォントデータ量が多く、全てのフォントデータを上記メモリに格納できない場合には、残りのフォントデータを継続して補助記憶装置に格納する手段とを設けたものである。

【0009】 また、複数の情報出力装置をエミュレートする手段を有する情報出力装置において、上位装置よりフォントデータをダウンロードする手段と、エミュレート可能な情報出力装置の間でダウンロードされたフォントデータを共有化する手段と、該手段によって共有化されたフォントデータを用いてエミュレートされた情報出

3

力装置の種類に係わらず文字イメージとしてビットマップメモリ上に展開する手段と、共有化されたフォントデータが存在するかどうかを判断する手段と、該手段によって共有化されたフォントデータが存在すると判断された場合には、該フォントデータが格納されたメモリエリアをエミュレーションの切り替えが行なわれても保持する手段とを設けたものも提供する。

【0010】なお、この情報出力装置において、共有化されたフォントデータが格納されたメモリエリアをエミュレーションの切り替えが行なわれた場合に保持するか否かを選択する手段を設けるとよい。

【0011】

【作用】第1の発明によれば、上位装置よりダウンロードするフォントデータを全て装置内のメモリに格納できない場合には、残りのフォントデータを継続して補助記憶装置に格納するので、装置内のメモリに必要なフォントデータがない場合、それを補助記憶装置からダウンロードすることにより、上位装置との印字データの通信を一時的に中断するようなことを回避でき、装置の使用効率が向上する。

【0012】また、第2の発明によれば、エミュレート可能な情報出力装置の間でダウンロードされたフォントデータを共有化し、その共有化されたフォントデータを用いてエミュレートされた情報出力装置の種類に係わらず文字イメージとしてビットマップメモリ上に展開することができる。また、共有化されたフォントデータが存在する場合には、そのフォントデータが格納されたメモリエリアをエミュレーションの切り替えが行なわれても保持するので、再び切り替え前のエミュレーションに戻した場合でも、そのフォントデータを上位装置から再度ダウンロードしないで済む。

【0013】なお、共有化されたフォントデータが格納されたメモリエリアをエミュレーションの切り替えが行なわれた場合に保持するか否かを選択する手段を設け、そのメモリエリアをエミュレーションの切り替えが行なわれた場合に保持しないようにした場合には、装置内のメモリをフルに使用できるので、ユーザにとってのメモリの使用方法の幅を広げることができる。

【0014】

【実施例】以下、この発明の実施例を図面に基づいて具体的に説明する。図2は、この発明をレーザプリンタに適用した一実施例のシステム構成図であり、1は上位装置であるパーソナルコンピュータ、ワードプロセッサ、データ処理装置、画像処理装置等のホストコンピュータ、2はページプリンタであるレーザプリンタ、3は補助記憶装置である。

【0015】レーザプリンタ2は、ホストコンピュータ1から送られてくる文字情報、イメージ(グラフィック)情報、あるいはベクトル情報等のプリントデータを解析し、ページ単位でビットマップデータ(ビデオデータ)

4

に変換するプリンタコントローラ4と、そのビットマップデータに応じてレーザを制御し、印字用紙に文字や画像等の可視画像をプリントするための作像部とそのシーケンスコントローラ等からなるプリンタエンジン5と、各種表示器、スイッチ等を備えた操作パネル6とによって構成されている。

【0016】図3は、プリンタコントローラ4の内部構成を示すブロック図である。このプリンタコントローラ4は、インタフェース(I/F)として、ホストコンピュータ1と接続するためのホストインタフェース11と、プリンタエンジン5と接続するためのエンジンインタフェース12と、操作パネル6と接続するためのパネルインタフェース13と、フロッピディスク装置14、読み書き可能なRAMカートリッジ15、読み出し専用のROMカートリッジ16等の補助記憶装置3と接続するための補助記憶装置インタフェース17とを備えている。

【0017】また、このプリンタコントローラ4は、マイクロコンピュータ(以下「CPU」と称す)18と、ROM19及びRAM20を備えている。CPU18は汎用の16又は32ビットのマイクロコンピュータであり、このプリンタコントローラ全体の統括制御を司る。ROM19はリードオンメモリであり、CPU18が動作するためのプログラムの他に常駐フォントデータ等を格納している。

【0018】RAM20は大容量のランダムアクセスメモリであり、主として次のような用途に使用される。

(a)インプットバッファ (b)ページバッファ (c)ビットマップメモリ  
(d)システムメモリ (e)フォントファイル (f)イメージファイル  
(g)ベクトルファイル

【0019】これ等のバッファ及びファイルは、システムソフトに使用する以外のRAM領域を必要な容量だけ事前に確保して使用する。この場合メモリ容量が不足した場合は、その旨を表示して機能を放棄する。その場合は、何等かのデータを消去して使用可能な状態に戻す必要がある。すなわち、全メモリ領域は固定的に割付けて使用するのではなく、必要な都度領域を確保して使用するダイナミック管理方式を採用している。

【0020】ここで、各バッファ及びファイルの機能について説明する。

〔インプットバッファ〕ホストインタフェース11に接続されているホストコンピュータ1からは、プリンタコントローラ4内の動作とは関係なく不定期にデータが送られてくる。したがって、ホストインタフェース11からのデータをインプットバッファに一時的に保持しておき、プリンタソフトウェアがそのデータを順次取り出して処理していく。

【0021】〔ページバッファ〕ホストコンピュータ1から送られてきたデータに、そのデータが実際に記憶さ

れているソースアドレスと、ビットマップメモリのどこにビットマップデータを作るかを指定するディスティネーションアドレス及びサイズ等の情報を付加して、このページバッファにページデータを作る。

【0022】【ビットマップメモリ】ビットマップメモリには、ページバッファのページデータと後述するフォントファイルのフォントデータ等、場合によりイメージファイルのイメージデータをも使用してビットマップデータが作成される。このビットマップメモリには、フル

ビットマップモードとパーシャルビットマップモードがあり、どちらのモードでも使用可能である。

【0023】フルビットマップモードは、ペーバサイズの全面に相当するメモリエリアをビットマップメモリとして割り付けて使用するモードであり、パーシャルビットマップモードは、ペーバサイズの一部分(例えば1/8ページ分)に相当するメモリエリアをビットマップメモリとして割り付けて、その半分ずつを交互に使用するモードである。

【0024】【フォントファイル】フォントデータはすべてファイルとして扱われており、ビットマップメモリにビットマップデータを展開する場合のソースデータとなる。したがって、使用可能なフォントデータが格納されているアドレスを示すリストと、各フォントデータ内の各文字のアドレスを示すリストがテーブルの形で設けられている。

【0025】フォントデータには、イメージフォントデータと圧縮フォントデータがあり、イメージフォントデータは矩形のドットマトリックス(ビットマップ形式)で表わされ、圧縮フォントデータは2桁のヘキサコードによる一連のデータの集まりで表現される。この実施例では、ホストコンピュータ1からダウンロードされるフォントデータはビットマップ形式に展開され、フォントファイルとしてRAM20に格納されるものとする。

【0026】【イメージファイル】ホストコンピュータ1から送られてくるイメージデータを順次格納し、イメージファイルを作る。ビットマップメモリがフルビットマップモードの場合には、このイメージファイルを作らずに直接ビットマップメモリにイメージデータを格納することもできる。パーシャルビットマップモードの場合は、イメージファイルとして使用可能なメモリ領域に作られる。そして、各イメージファイルには識別コードが付けられ、ページバッファに組み込まれる。

【0027】【ベクトルファイル】ホストコンピュータ1から送られてきたデータがベクトル情報である場合は、そのベクトルが作成されて、そのアドレスがページバッファに組み込まれる。パーシャルビットマップモードの場合で、ベクトル情報が複数のページブロックにまたがる場合は、ページブロック単位に分解されてファイルが作成される。フルビットマップモードの場合でも、小単位に分解される方がメモリの使用効率及び処理速度

上都合がよい。

【0028】次に、このように構成したこの実施例の作用について、図1も参照して具体的に説明する。図1は、図3のCPU18によるこの発明に係わる処理を示すフローチャートである。

【0029】このルーチンはホストコンピュータ1よりフォントデータがダウンロードされるとスタートし、そのフォントデータを順次ビットマップ形式に展開し、フォントファイルとしてRAM(主記憶装置)20に格納していく。そして、メモリ容量不足が生じる前にダウンロードが終了すれば処理を終了するが、ダウンロードが終了する前にメモリ容量不足が生じ、メモリオーバフローを検出した場合には、ホストコンピュータ1にダウンロードを中断させてその途中のフォントデータを消去する。

【0030】次いで、補助記憶装置3のメモリ容量をチェックし、そこに空があれば、ホストコンピュータ1にダウンロード途中で消去されたフォントデータの先頭より再びダウンロードを開始させ、それを補助記憶装置3に格納するが、ここでもメモリオーバフローになると、上述と同様にダウンロード途中のフォントデータを削除してダウンロードを終了する。

【0031】このように、この実施例によれば、ホストコンピュータ1よりダウンロードするフォントデータを全てRAM20に格納できない場合には、残りのフォントデータを継続して補助記憶装置3に格納するので、RAM20に必要なフォントデータがない場合、それを補助記憶装置3からダウンロードすることにより、ホストコンピュータ1との印字データの通信を一時的に中断するようなことがなくなり、装置の使用効率が向上する。

【0032】なお、画像処理過程において、ダウンロードしたフォントデータが格納されているRAM20のメモリ領域が必要となった場合には、一時的に必要な数のフォントデータを補助記憶装置3のメモリに移すことも可能である。

【0033】図4はこの発明の他の実施例を示すプリンタコントローラのブロック構成図である。このプリンタコントローラ30は、ホストコンピュータ31と接続するためのホストインタフェース32、プリンタエンジン33と接続するためのエンジンインタフェース34、操作パネル35と接続するためのパネルインタフェース36、オプションRAM37と接続するためのオプションRAMインタフェース38、ROMカートリッジ39と接続するためのROMカートリッジインタフェース40の各インタフェース回路と、マイクロコンピュータ(以下「CPU」と称す)41、ROM42、RAM43及びEEPROM44とを備えている。

【0034】CPU41は、ROM42内の制御プログラム及びEEPROM44内のデータに従って、ホストコンピュータ31から送信されるデータの処理及びその

7

他の制御を行なう。ROM42は、プリンタコントローラ30の制御プログラム及びプリントする際のフォントのパターン情報などを記憶している。

【0035】RAM43は、CPU41が使用するワークメモリ、ホストコンピュータ31からのデータを一時的に格納するインプットバッファ、プリンタエンジン33に出力するビットマップデータを格納するビットマップメモリ等に用いられる。EEPROM44は、操作パネル35からのモードセットの内容などを記憶しておく不揮発性メモリである。

【0036】オプションRAM37は、ビットマップデータ及びダウンロードフォントを一時的に格納するRAM51の増設用RAMである。ROMカートリッジ39は、プリンタコントローラ30に内蔵されているフォント以外のフォントを使ってプリントしたいときに使用するフォントカートリッジ、あるいはプリンタに他の機能（他のプリンタのエミュレーション）をさせたいときに使用するエミュレーションカートリッジである。

【0037】図5はプリンタコントローラ30のモジュール構成を示す説明図である。スーパーユーザは、全てのエミュレーションで共通の機能を司るモジュール群で成り立ち、エミュレーションは各エミュレーション独自の機能を司るモジュール群で成り立っている。スーパーユーザは、各インタフェースの制御、及びインプットバッファの取り扱い、ビットマップメモリに対するドロー（描

```

ESC [m1, m2@ font-name
ESC [header
ESC SP ESC SP
ESC 'm11, m12, m13, m14, m15, m16, m17@イメージデータ1
ESC 'm21, m22, m23, m24, m25, m26, m27@イメージデータ2
. . . . .
ESC 'mk1, mk2, mk3, mk4, mk5, mk6, mk7@イメージデータk
ESC ]

```

【0042】ここで、m1はイメージデータを並べるフォーマットを決定するパラメータである。m2によりこのダウンロードフォントがポートレートまたはランドスケープ印字用のフォントであるか否かが決定される。

【0043】次のフォントヘッダ（header）はそのダウンロードフォントの属性を示すパラメータの集合体である。その中にはそのダウンロードフォントがイタリックであるか、ボールドであるか、あるいは固定ピッチのフォントか、プロポーショナルフォントか、デフォルトHMI、VMIの値がなんであるかなどの様々なフォント情報が含まれている。このフォントヘッダの具体的な属性の内容の一例を図6に示す。

【0044】また、mk1はASCIIコードナンバを示し、mk2～mk7は図7にも示すようにそれぞれイメージの実水平幅、文字幅の左端からイメージまでのオフセット（セルからのレフトオフセット）、文字幅、イ

8

\*画）の制御、ジャムバックアップなどの機能を持つ。

【0038】エミュレーションは、独自のフォントのハンドリング、独自コマンドのインタプリタなどからなる。ダウンロードフォントを扱うモジュールはエミュレーション部分に属し、個々のエミュレーションによって処理形式が異なっている。

【0039】エミュレーション1にはコマンド1のフォーマットで定義されたダウンロードフォントが送られ、エミュレーション1のインタプリタ部分でスーパーユーザに解釈可能なように共通フォーマットへの変換が行なわれる。スーパーユーザでは、この変換された共通フォーマットをもとにビットマップメモリでのドローが行なわれた後、プリント要求がプリンタエンジン33に対して出力される。

【0040】エミュレーション2でもコマンド2のフォーマットで定義されたダウンロードフォントが共通フォーマットに変換されて全く同様の動作を繰り返す。また、複数のエミュレーションがある場合には、そのエミュレーションの数だけフォントをダウンロードする形式が存在するが、全てインタプリタ部分で共通フォーマットに変換される。

【0041】以下に、フォントをダウンロードするときのエミュレーション1での具体的なコマンドフォーマットを示す。

メージの実垂直高、文字高の上端からイメージまでのオフセット（セルからのトップオフセット）、文字の高さを示している。

【0045】図8は先のフォントヘッダがフォント全体の情報を受渡したのとは異なり、個々の文字のキャラクタヘッダの内容を示し、個々の文字の特徴を表すような上述したmk1～mk7の各情報が含まれている。このキャラクタヘッダは、ダウンロードする文字数分存在し、このヘッダのすぐ後に実際のイメージ情報が指定されたフォーマットに従って送られる。

【0046】以下に、フォントをダウンロードするときのエミュレーション2での具体的なコマンドフォーマットを示す。

```

ESC * c m1 D
ESC ) s m2 W      (フォントディスクリプタ)
ESC * c m3 E

```

ESC (s m4 W (キャラクタディスクリプタ  
&イメージデータ)

【0047】ここで、m1はIDナンバ、m2はフォントヘッダのバイト数、m3はASCII10進、m4はイメージ&データのバイト数をそれぞれ示している。このコマンドは上述したエミュレーション1でのコマンドと明らかに始めのエスケープシーケンスのところから異なっており、フォントヘッダにあたる部分はフォントディスクリプタと呼ばれている。

【0048】フォントディスクリプタには、図9に示すように先に上げたフォントヘッダとほぼ同様の内容が含まれている。また、エミュレーション1でのキャラクタヘッダにあたるものは図10に示すキャラクタディスクリプタであり、これも同様にかなり類似した内容になっている。なお、図10のレフトオフセットは前の印字位置からイメージのドローを開始する位置までのオフセットを、トップオフセットはイメージのトップからベースラインまでのオフセットを、デルタXは前の印字位置から次の印字位置までの長さ(ドット)をそれぞれ表わす。

【0049】一般には図11に示すように、まずエスケープシーケンスを送り、次にフォントの共通情報の入ったフォントヘッダを送り、次に個々のコードに対するキャラクタヘッダ情報とイメージデータをコード数分送る。これらのフォント情報はエミュレーション1及びエミュレーション2の中でそれぞれスーパユーザが解釈できるように各エミュレーション共通のフォーマットに変換される。

【0050】この実施例では、この共通フォーマットをある特定の領域に格納し、エミュレーションが変更されてもこの特定領域についてのみメモリのクリヤが行なわれないように制御することによって、複数のエミュレーションが存在するプリンタ内でエミュレーションの切り替えが行なわれたとしてもダウンロードフォントが共通に使えるようになる。

【0051】図12、図13、図14に、それぞれファイルヘッダ、フォントヘッダ、キャラクタヘッダの各エミュレーションによって変換された共通フォーマットの具体例を示す。なお、図14のdwはイメージの主走査方向の幅、dhはイメージの副走査方向の幅、lwはイメージの幅、lhはイメージの高さ、cwはセルの幅、dxはアクティブポジションの移動量をそれぞれ表わす。

【0052】また、図15にRAM43内のインプットバッファ、変数エリア、ビットマップエリア、ダウンロードフォント格納エリアをそれぞれ示す。なお、そのダウンロードフォント格納エリアに上記共通フォーマットが格納される。

【0053】図16はこの実施例におけるCPU41によるこの発明に係わる処理を示すフローチャートであ

り、まずエミュレーション変更の指示がなされたかどうかを判断し、エミュレーション変更の指示がなされた場合、共有化されたダウンロードフォント(共通フォーマット)がRAM43内に存在するか否かを判断する。

【0054】そして、RAM43内に共有化されたダウンロードフォントが存在しない場合には、RAM43内の全てのエリア(データ)をクリヤした後、エミュレーションの切り替えを行ない、共有化されたダウンロードフォントが存在する場合には、RAM43内のダウンロードフォント格納エリア以外のエリアをクリヤした後、エミュレーションの切り替えを行なう。

【0055】このように、この実施例によれば、エミュレート可能なプリンタの間でダウンロードされたフォントデータを共有化し、その共有化されたフォントデータが格納されたメモリエリアをエミュレーションの切り替えが行なわれても保持するので、切り替え前のエミュレーションに戻した場合でも、そのフォントデータをホストコンピュータ31から再度ダウンロードしないで済む。

【0056】図17は、CPU41によるこの発明に係わる図16と異なる処理を示すフローチャートであり、まずエミュレーション変更の指示がなされたかどうかを判断し、エミュレーション変更の指示がなされると、操作パネル35からの操作信号又はホストコンピュータ31からのコマンドによってダウンロードフォント共通使用モード(RAM43内のダウンロードフォント格納エリアをクリヤしないモード)が設定されているかどうかを判断する。

【0057】そして、ダウンロードフォント共通使用モードが設定されていない場合には、RAM43内の全てのエリアをクリヤした後、エミュレーションの切り替えを行ない、ダウンロードフォント共通使用モードが設定されている場合には、共有化されたダウンロードフォントがRAM43内に存在するか否かを判断し、存在しない場合にも、RAM43内の全てのエリアをクリヤした後、エミュレーションの切り替えを行なう。

【0058】また、共有化されたダウンロードフォントが存在する場合には、RAM43内のダウンロードフォント格納エリア以外のエリアをクリヤした後、エミュレーションの切り替えを行なう。

【0059】この処理によれば、ユーザがダウンロードフォント共通使用モードを選択しない場合、RAM43内のダウンロードフォント格納エリアもクリヤされるが、そうすることによってRAM43をフルに使用することができ、ユーザにとってのメモリの使用幅を広げることができる。

【0060】以上、この発明をレーザプリンタに適用した実施例について説明したが、この発明はこれに限らず、LEDプリンタ等の他のページプリンタは勿論、デジタル複写機、普通紙ファックス装置等の各種画像形成

装置、さらにはCRTディスプレイ装置などの各種の情報処理装置に適用可能である。

【0061】

【発明の効果】以上説明してきたように、この発明によれば、上位装置よりフォントデータをダウンロードする機会が相当減るので、装置全体の使用効率が向上する。

【図面の簡単な説明】

【図1】図3のCPU18によるこの発明に係わる処理を示すフロー図である。

【図2】この発明をレーザプリンタに適用した一実施例のシステム構成図である。

【図3】図2のプリンタコントローラの内部構成を示すブロック図である。

【図4】この発明の他の実施例であるプリンタコントローラの内部構成を示すブロック図である。

【図5】図4のプリンタコントローラのモジュール構成を示す説明図である。

【図6】図5のエミュレーション1に対応するフォントヘッダの一例を示す説明図である。

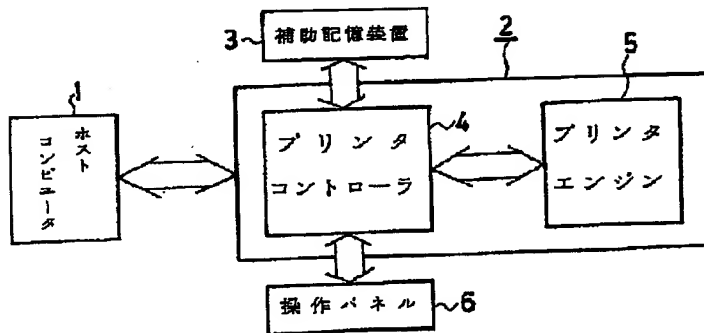
【図7】同じくエミュレーション1に対応するフォントをダウンロードするためのコマンドを説明するための説明図である。

【図8】同じくエミュレーション1に対応する個々の文字のキャラクタヘッダの一例を示す説明図である。

【図9】同じくエミュレーション2に対応するフォントディスクリプタの一例を示す説明図である。

【図10】同じくエミュレーション2に対応する個々の文字のキャラクタディスクリプタの一例を示す説明図である。

【図2】



ある。

【図11】フォントをダウンロードする際の一般的なコマンドフォーマットを示す説明図である。

【図12】図5の各エミュレーションによって変換されたファイルヘッダの共通フォーマットの一例を示す説明図である。

【図13】同じく各エミュレーションによって変換されたフォントヘッダの共通フォーマットの一例を示す説明図である。

【図14】同じく各エミュレーションによって変換されたキャラクタヘッダの共通フォーマットの一例を示す説明図である。

【図15】図4のRAM43内のメモリエリアを示すメモリマップ図である。

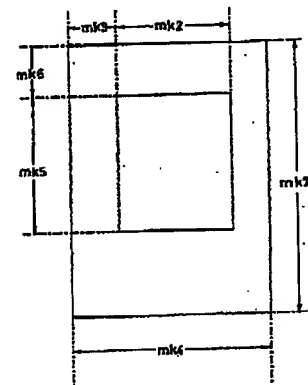
【図16】図4のCPU41によるこの発明に係わる処理を示すフロー図である。

【図17】図4のCPU41によるこの発明に係わる図16と異なる処理を示すフロー図である。

【符号の説明】

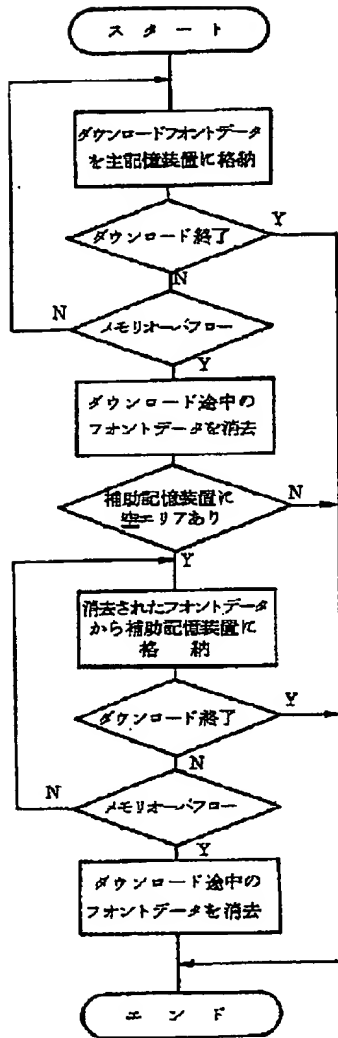
1, 31	ホストコンピュータ	2	レーザプリンタ
3	補助記憶装置	4, 30	プリンタコントローラ
5, 33	プリンタエンジン	6, 35	操作パネル
14	フロッピーディスクリッジ	15	RAMカートリッジ
18, 41	CPU	19, 42	ROM
20, 43	RAM	44	EEPROM

【図7】

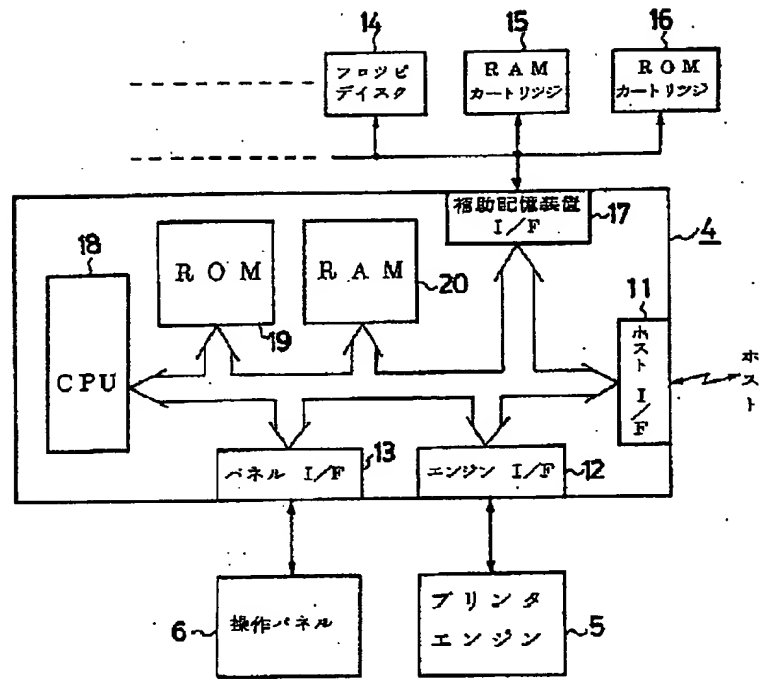




【図1】



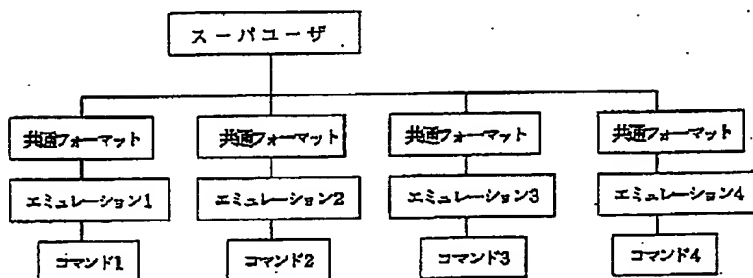
【図3】



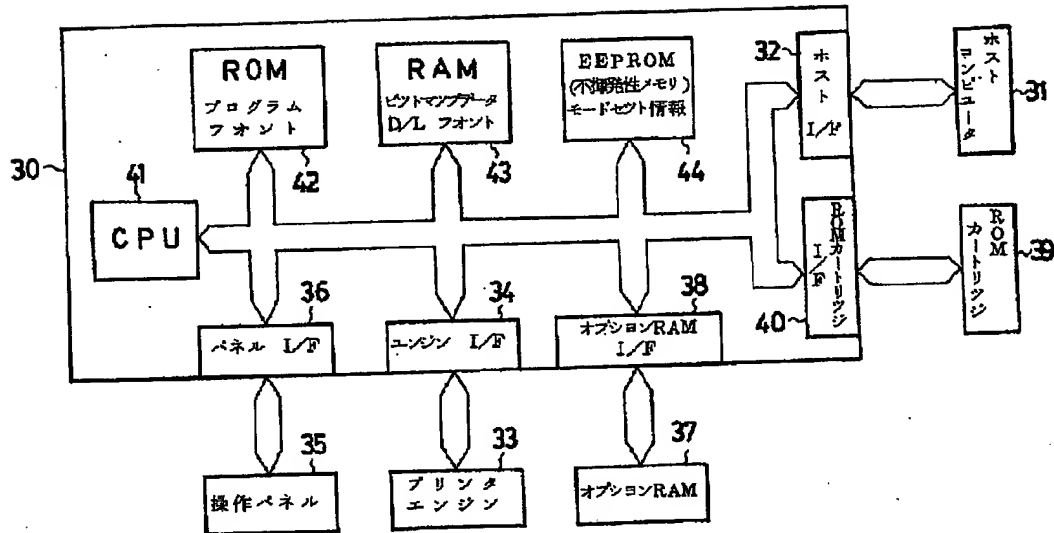
【図8】

ASCIIコード
イメージの奥行き幅
レフトオフセット
文字幅
イメージの実用高さ
トップオフセット
文字の高さ

【図5】



【図4】



【図9】

フォント・ディスクリプタのサイズ
ベースラインオフセット
セル幅
セルの高さ
フォントタイプ
オリエンテーション
スペーシング
シンボルセット
デフォルトHMI
デフォルトVMI
X ハイット
プロポーショナルHMI
スタイル
ストローク・ウエイト
タイプフェイス
アンダ・ラインオフセット
アンダ・ラインスリックネス

【図10】

イメージ描画フォーマット
ディスクリプタサイズ
オリエンテーション
レフト・オフセット
トップ・オフセット
イメージ幅
イメージ高さ
デルタ X
イメージデータ

【図11】

ESCシーケンス
共通フォント情報
個々のフォント情報(1)
イメージデータ(1)
個々のフォント情報(2)
イメージデータ(2)
.....

【図12】

(フォントヘッダ)  
 (個々のコードに付するヘッダ 情報)  
 (イメージデータ)

【図14】

ファイルサイズ
CRC
ファイルタイプ
ヘッダオフセット
ファイルネーム
ファイルバージョン
ファイルデータ
ファイル・コピーライト

イメージオフセット
dw
dh
レフトオフセット
トップオフセット
!w
!h
cw
dx

【図6】

バージョン
フォントネーム
フォントコピーライト
水平方向解像度
垂直方向解像度
デフォルトHMI
デフォルトVMI
IDナンバー
ファーストIDナンバー
ラストIDナンバー
ローテーション
キャラクター セットID
プロポーショナルモード
イタリック
ボールド
ベースオフセット
アング スコア オフセット
アング スコア スティックネス
シャドウ印刷体水平オフセット
シャドウ印刷体垂直オフセット
ボールド印刷体水平オフセット
ボールド印刷体垂直オフセット

【図13】

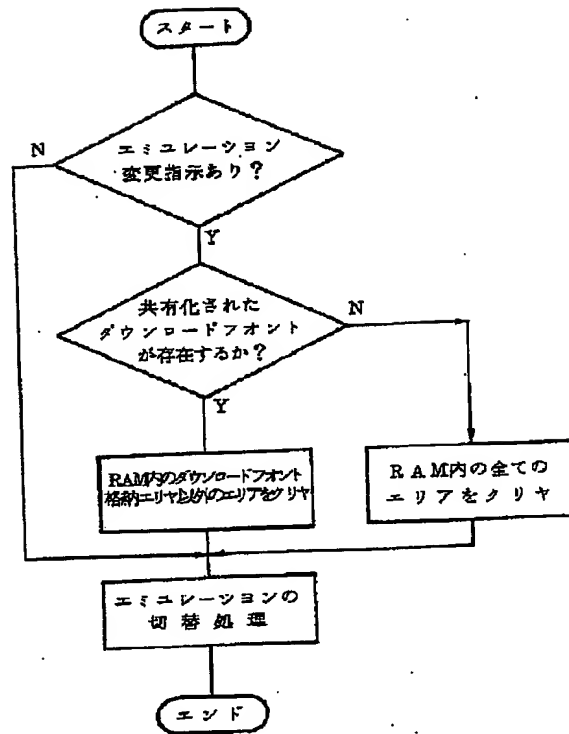
フォントタイプ
プールID
フォントネーム
デフォルトアサイン
ファーストグリフID
ラストグリフID
ローテーション
シンボルセット
スペーシング
セル幅
セル高
HMI
VMI
タイプサイズ
スタイル
ウェイト
タイプフェイス
ベースオフセット
スペース幅
アング ラインオフセット
アング ライン幅
シャドウ印刷体水平オフセット
シャドウ印刷体垂直オフセット
ボールド印刷体水平オフセット
ボールド印刷体垂直オフセット

【図15】

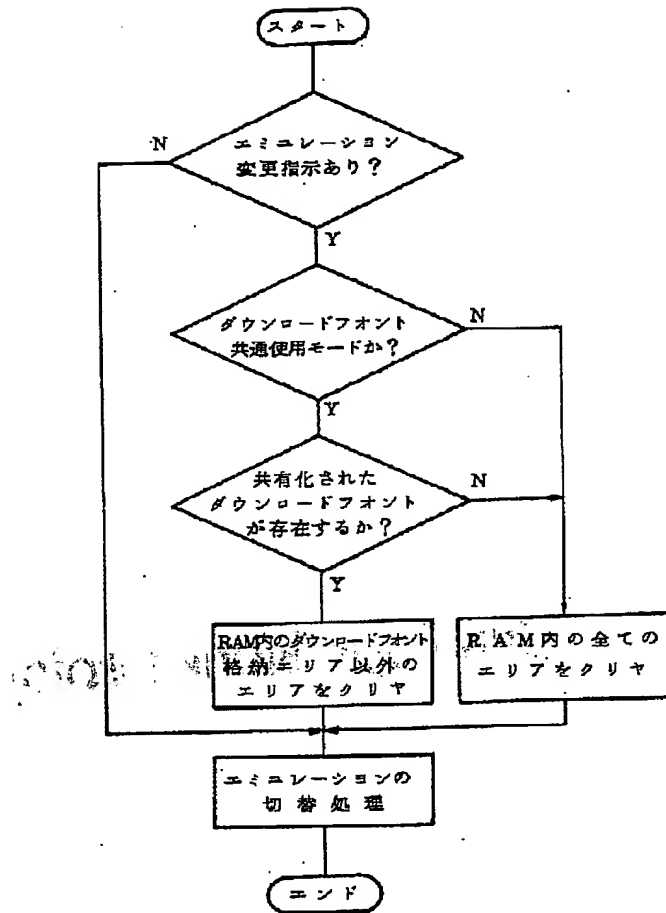
インプットマップ
変換エリア
ビットマップエリア
ダウンロードフォント格納 エリア

43

【図16】



【図17】



***This Page Blank (uspto)***